CLAIM AMENDMENTS

1. (Previously Presented): A method of streaming data over a network from a first device to a second device, the method comprising:

compressing the data at the first device by finding an index in a lookup table that matches an initial sequence in the data, the lookup table comprising a plurality of entries, each said entry being discoverable utilizing a particular one of a plurality of said indices, each said entry referencing whether a corresponding said index is located in a history buffer, and if so, further references one or more locations of the corresponding said index in the history buffer;

when the corresponding said entry of the matching index references a plurality of said locations:

for each said location, comparing a sequence at the location having the matching index with a sequence in the data, said sequence including the initial sequence;

deriving a matching sequence from the comparison based on at least one of a length and the location of the sequence at each said location;

representing the matching sequence using a representation that includes the length and the location of the matching sequence in the history buffer;

forming compressed data that includes at least one of said representations;

further compressing the compressed data by encoding the at least one representation that includes the length and the location of the matching sequence, the at least one representation being encoded using a first Huffman table for encoding the length using Huffman encoding;

using a last recently used (LRU) table for encoding the location of the matching sequence in the history buffer, the LRU table listing a plurality of recently used locations of recent matching sequences;

when the location of the matching sequence is not in the LRU table, encoding the location of the matching sequence with Huffman encoding using a second Huffman table, different from the first Huffman table; and

streaming the compressed data over the network to the second device.

2. (Previously Presented): A method as described in claim 1, wherein the forming compressed data includes finding one said index in the lookup table for each said sequence in the data.

3. (Original): A method as described in claim 1, wherein the corresponding said entry of the matching index references a hash chain which includes each said location of the matching index in the history buffer.

4. (Original): A method as described in claim 1, wherein the initial sequence and the index are each composed of at least two bytes.

5. (Previously Presented): A method as described in claim 1, further comprising streaming the compressed data over a network, wherein the data is formatted as one or more packets and the packets are compressed for transmission over the network so that the compressing is performed on a per-packet basis.

6. (Previously Presented): A method as described in claim 1, further comprising:

using the second Huffman table to also compress literal sequences that have no matching index in the history buffer; and

streaming the compressed literal sequences to the second device.

7. (Previously Presented): A method as described in claim 1, further comprising:

determining that the corresponding said entry of the matching index references a single said location;

comparing a sequence at the single said location having the matching index with the sequence in the data;

deriving a matching sequence from the comparison based on at least one of a length and the location of the sequence at the single said location; and

representing the matching sequence using a representation that includes the length and the single said location of the matching sequence in the history buffer; and

when each said sequence of the data is represented or encoded, streaming the data having the encoding or the representation.

8. (Original): A method as described in claim 1, wherein the comparison to derive the matching sequence is performed utilizing one or more thresholds selected from the group consisting of:

a number of said locations having the matching index to be compared;

a size of a value that describes each said location having the matching index; and

a size of a value that describes a length of the sequence at each said location that matches the sequence in the data that includes the matching index.

9. (Original): A method as described in claim 1, further comprising employing a cost function to determine if the representation utilizes less memory when stored than the matching sequence, and if so, forming compressed data that includes the representation.

10. (Previously Presented): A method as described in claim 1, further comprising determining whether the location of the matching sequence matches one of a plurality of locations in the LRU table, wherein:

each said location in the LRU table has a corresponding said LRU representation;

each said location in the LRU table describes one of a plurality of last recently used locations of sequences in previously streamed data; and

if the location of the matching sequence is included in the LRU table, the location of the matching sequence is encoded with a corresponding said LRU representation from the LRU table.

11. (Previously Presented): One or more computer-readable storage media storing computer-executable instructions that, when executed, perform the method as recited in claim 1.

12. (Currently Amended): A method ~~of compressing data, the method~~ comprising:

adding data to a history buffer;

updating a lookup table that references the history buffer to include the added data, the lookup table comprising a plurality of entries, each said entry being discoverable utilizing a particular one of a plurality of indices, each said entry referencing whether a corresponding said index is located in a history buffer, and if so, further referencing one or more locations of the corresponding said index in the history buffer;

starting a current pointer at the added data in the history buffer;

finding one said index in the lookup table that matches an initial sequence at the current pointer;

determining that the corresponding said entry of the matching index references a plurality of said locations;

comparing a sequence at each said location having the matching index with a sequence in the added input data that includes the initial sequence;

deriving a matching sequence from the comparison;

representing the matching sequence with a representation that includes the location and a length of the matching sequence in the history buffer;

~~employing a cost function to determine that the representation utilizes less memory space when stored than the matching sequence;~~

configuring data to include the representation, advancing the current pointer by the length of the matching sequence, and encoding at least a portion of the representation to further compress the data by encoding the representation, the representation being encoded using a first Huffman table for encoding the length using Huffman encoding;

using a last recently used (LRU) table for encoding the location of the matching sequence in the history buffer, the LRU table listing a plurality of recently used locations of recent matching sequences;

when the location of the matching sequence is not in the LRU table, encoding the location of the matching sequence with Huffman encoding using a second Huffman table, different from the first Huffman table; ~~and~~

packetizing the configured data for streaming; and

streaming the configured data over a network.

13. (Previously Presented): A method as described in claim 12, further comprising

using the second Huffman table to also compress literal sequences that do not match any sequence in the history buffer; and

streaming the compressed literal sequences.

14. (Original): A method as described in claim 12, wherein when the initial sequence at the current pointer does not match any sequence in the history buffer, the initial byte sequence at the current pointer is encoded for inclusion into configured data for streaming by an encoding technique selected from the group consisting of:

Huffman encoding;

arithmetic encoding;

prefix encoding; and

Markov encoding.

15. (Previously Presented): A method as described in claim 12, further comprising determining whether the location of the matching sequence matches one of the plurality of recently used locations in the LRU table, wherein:

each said location in the LRU table has a corresponding said LRU representation;

each said location in the LRU table describes one of a plurality of last recently used locations of sequences in previously streamed data; and

if the location of the matching sequence is included in the LRU table, the location of the matching sequence is encoded with a corresponding said LRU representation from the LRU table.

16. (Previously Presented): One or more computer-readable storage media storing computer-executable instructions that, when executed, perform the method as recited in claim 12.

17. (Currently Amended): A method for streaming ~~transmitting compressed~~ data over a network from a first device to a second device ~~using a compression process~~, the method comprising:

adding data to a history buffer at the first device for compression;

updating a lookup table that references the history buffer to include the added data, the lookup table comprising a plurality of entries, each said entry being discoverable utilizing a particular one of a plurality of indices, each said entry referencing whether a corresponding said index is located in the history buffer, and if so, further referencing one or more locations of the corresponding said index in the history buffer;

starting a current pointer at the added data in the history buffer;

finding one said index in the lookup table that matches an initial sequence at the current pointer;

determining that the corresponding said entry of the matching index references a plurality of said locations;

comparing a sequence at each said location having the matching index with a sequence in the added data that includes the initial sequence;

deriving a matching sequence from the comparison;

representing the matching sequence with a representation that includes the location and a length of the matching sequence in the history buffer;

forming compressed data from the packet of data that includes the representation;

advancing the current pointer by the length of the matching sequence;

using Huffman encoding to compress literal sequences that have no matching index in the history buffer, the Huffman encoding using a frequency of occurrences table generated from the data and a variable length string assigned as a prefix to each literal sequence for uniquely representing the literal sequence;

when the current pointer has advanced through the packet of data, packetizing the compressed data for streaming; and

streaming the packetized compressed data over the network to the second device.

18. (Previously Presented): A method as described in claim 17, further comprising encoding the representation to further compress the compressed data, wherein the representation is encoded using a first Huffman table for encoding the length using Huffman encoding and using a last recently used (LRU) table for encoding the location of the matching sequence in the history buffer, wherein the LRU table lists a plurality of recently used locations of recent matching sequences, wherein, when the location of the matching sequence is not in the LRU table, the location of the matching sequence is encoded with Huffman encoding using a second Huffman table.

19. (Previously Presented): One or more computer-readable storage media storing computer-executable instructions that, when executed, perform the method as recited in claim 17.

20. - 22. (Canceled)

23. (Currently Amended): A server comprising:

a processor and a memory, the memory including a compression module executed by the server for implementing:

a history buffer having a plurality of bytes;

a lookup table that includes a plurality of entries, each said entry:

being discoverable utilizing a particular one of a plurality of indices; and

references whether a corresponding said index is located in the history buffer, and if so, further references one or more locations of the corresponding said index in the history buffer; and

the compression module being executable to:

find one said index sequence in the lookup table that matches an initial sequence in data for communication to a client from a terminal service;

determine that the corresponding said entry of the matching index references a plurality of said locations;

for each said location, compare a sequence at the location having the matching index with a sequence in the data, said sequence including the initial sequence;

derive a matching sequence from the comparison based on at least one of a length and the location of the sequence at each said location;

represent the matching sequence using a representation that includes the length and the location of the matching sequence in the history buffer;

compress at least a portion of the representation by encoding the representation, the representation being encoded using a first Huffman table for encoding the length using Huffman encoding;

~~using~~ use a last recently used (LRU) table for encoding the location of the matching sequence in the history buffer, the LRU table listing a plurality of recently used locations of recent matching sequences;

when the location of the matching sequence is not in the LRU table, encode the location of the matching sequence with Huffman encoding using a second Huffman table, different from the first Huffman table; ~~and~~

~~using~~ use Huffman encoding to compress literal sequences that have no matching index in the history buffer, the Huffman encoding using a frequency of occurrences table generated from the data and a variable length string assigned as a prefix to each literal sequence for uniquely representing the literal sequence;

form compressed data that includes one or more said representations; and

stream the compressed data over a network.

24. (Currently Amended): A server as described in claim 23, wherein the compression module is further executable to[[:]] find one said index in the lookup table for each said sequence in the data[[;]] ~~form compressed data that includes one or more said representations; and stream the compressed data~~.

25. (Original): A server as described in claim 23, wherein the corresponding said entry of the matching index references a hash chain which includes each said location of the matching index in the history buffer.

26. (Original): A server as described in claim 23, wherein the initial sequence and the index are each composed of at least two bytes.

27. (Currently Amended): A server as described in claim 23, wherein the compression module is further executable to[[:]] ~~form compressed data that includes the representation; and~~ packetize the compressed data for streaming over a network, wherein the data is formatted as one or more packets and compressed on a per-packet basis.

28. (Canceled)

29. (Previously Presented): A server as described in claim 23, wherein the compression module is further executable to periodically recalculate the first Huffman table and the second Huffman table following processing of a predetermined number of packets of the data.

30. (Original): A server as described in claim 23, wherein the comparison to derive the matching sequence is performed utilizing one or more thresholds selected from the group consisting of:

a number of said locations having the matching index to be compared;

a size of a value that describes each said location having the matching index; and

a size of a value that describes a length of the sequence at each said location that matches the sequence in the data that includes the matching index.

31. (Original): A server as described in claim 23, wherein the compression module is further executable to employ a cost function to determine if the representation utilizes less memory when stored than the matching sequence, and if so, forming compressed data that includes the representation.

32. (Previously Presented): A server as described in claim 23, wherein the compression module is further executable to determine whether the location of the matching sequence matches one of the plurality of locations in the LRU table, wherein:

each said location in the LRU table has a corresponding said LRU representation;

each said location in the LRU table describes one of a plurality of last recently used locations of sequences in previously streamed data; and

if the location of the matching sequence is included in the LRU table, the location of the matching sequence is encoded with a corresponding said LRU representation from the LRU table.

33. (Currently Amended): A system for streaming data, the system comprising:

a server including a first processor and a first memory and further comprising:

a first history buffer having a plurality of bytes;

a lookup table that includes a plurality of entries, each said entry being discoverable utilizing a particular one of a plurality of indices, each said entry referencing whether a corresponding said index is located in the history buffer, and if so, one or more locations of the corresponding said index in the history buffer;

a first Huffman table that includes codes for lengths of matching sequences;

a second Huffman table, different from the first Huffman table, that includes codes for locations of matching sequences and literal bytes; and

a compression module that is executable by the server to:

find one said index in the lookup table that matches an initial sequence at a current pointer in data to be streamed in response to a request for remote access;

if the corresponding said entry of the matching index references one or more said locations:

compare a sequence at each said location having the matching index with a sequence in the data at the current pointer;

derive a matching sequence from the comparison;

configure data to include a representation that includes the location and a length of the matching sequence in the first history buffer,

encode the length using the first Huffman table,

encode the location of the matching sequence using a last recently used (LRU) table, the LRU table listing a plurality of recently used locations of recent matching sequences;

when the location of the matching sequence is not in the LRU table, the location of the matching sequence is encoded using the second Huffman table, and advance the current pointer by the length of the matching sequence;

if the corresponding said entry of the matching index does not reference any said location, configure data to include the initial sequence, encode literal bytes of the initial sequence using the second Huffman table and a variable length string assigned as a prefix to each literal byte for uniquely representing the literal byte, and advance the current pointer by a length of the initial sequence; and

when the current pointer has advanced through the added data, further configure the data into packets, and stream the packetized configured data over a network; and

a client including a second processor and a second memory, the client being communicatively coupled to the network for communication with the server and including a second said history buffer, a third Huffman table that includes codes for decoding locations of matching sequences and literal bytes, a fourth Huffman table that includes codes for decoding lengths of matching sequences, the LRU table, and a decompression module that is executable by the client to decompress the streamed data;

when an encoded representation is present in the configured data, the decompression module being configured to decode the representation using the LRU table, the third Huffman table, and the fourth Huffman table, and find the matching sequence in the second said history buffer based on a decoded location and a decoded length indicated by the representation.

34. (Original): A system as described in claim 33, wherein the decompression module is further executable by the client to add decompressed data to the second history buffer.

35. (Previously Presented): A system as described in claim 33, wherein the server is further configured to:

receive feedback that indicates availability of resources for communicating the packetized configured data over the network from the server to the client; and

tune one or more parameters of the compression module in response to the feedback.

36. (Previously Presented): A system as described in claim 35, wherein tuning the one or more parameters further comprises increasing a size of a search window used for sequence matching by the compression module when the feedback indicates that data is being transmitted over the network at a lower than expected rate.

37. (Previously Presented): A system as described in claim 33, wherein the compression module is further executable by the server to determine whether the location of the matching sequence matches the plurality of locations in the LRU table, wherein:

each said location in the LRU table has a corresponding said LRU representation;

each said location in the LRU table describes one of a plurality of last recently used locations of sequences in previously streamed data; and

if the location of the matching sequence is included in the LRU table, the location of the matching sequence is encoded with a corresponding said LRU representation from the LRU table.

38. (Currently Amended): A computer-readable storage medium storing computer-executable instructions that, when executed by a computer, direct the computer to:

find an index in a lookup table that matches an initial sequence in data for streaming to a client, the data for generating a user interface of an application that is being executed remotely from the client, the lookup table comprising a plurality of entries, each said entry being discoverable utilizing a particular one of a plurality of said indices; and each said entry references whether a corresponding said index is located in a history buffer; and if so, further references one or more locations of the corresponding said index in the history buffer;

determine that the corresponding said entry of the matching index references a plurality of said locations;

for each said location, compare a sequence at the location having the matching index with a sequence in the data, said sequence including the initial sequence;

compute, from the comparison, a length of the matching sequence;

represent the matching sequence using a representation that includes the length and the location;

compress the representation by encoding the representation, the representation being encoded using a first Huffman table for encoding the length using Huffman encoding;

use a last recently used (LRU) table for encoding the location of the matching sequence in the history buffer, the LRU table listing a plurality of recently used locations of recent matching sequences;

when the location of the matching sequence is not in the LRU table, encode the location of the matching sequence with Huffman encoding using a second Huffman table, different from the first Huffman table; ~~and~~

use Huffman encoding to compress literal sequences that have no matching index in the history buffer, the Huffman encoding using a frequency of occurrences table generated from the data and a variable length string assigned as a prefix to each literal sequence for uniquely representing the literal sequence; and

stream the data to the client.

39. (Canceled)

40. (Previously Presented): A computer-readable storage medium as described in claim 38, wherein the corresponding said entry of the matching index references a hash chain which includes each said location of the matching index in the history buffer.

41. (Previously Presented): A computer-readable storage medium as described in claim 38, wherein the initial sequence and the index are each composed of at least two bytes.

42. (Previously Presented): A computer-readable storage medium as described in claim 38, wherein the computer-executable instructions direct the computer to receive feedback that indicates availability of resources for communicating the packetized configured data over the network from the server to the client; and

tune one or more parameters of the compression module in response to the feedback.

43. (Previously Presented): A computer-readable storage medium as described in claim 38, wherein the computer-executable instructions direct the computer to periodically recompute the first Huffman table and the second Huffman table following processing of a predetermined number of packets of the data.

44. (Previously Presented): A computer-readable storage medium as described in claim 38, wherein the comparison is performed utilizing one or more thresholds selected from the group consisting of:

a number of said locations having the matching index to be compared;

a size of a value that describes each said location having the matching index; and

a size of a value that describes a length of the sequence at each said location that matches the sequence in the data that includes the matching index.

45. (Previously Presented): A computer-readable storage medium as described in claim 38, wherein the computer-executable instructions direct the computer to employ a cost function to determine if a representation that includes the length and the location of the matching sequence utilizes less memory when stored than the matching sequence, and if so, form compressed data that includes the representation.

46. (Previously Presented): A computer-readable storage medium as described in claim 38, wherein the computer-executable instructions direct the computer to determine whether the location of the matching sequence matches one of the plurality of locations in the LRU table, wherein:

each said location in the LRU table has a corresponding said LRU representation;

each said location in the LRU table describes one of a plurality of last recently used locations of sequences in previously streamed data; and

if the location of the matching sequence is included in the LRU table, the location of the matching sequence is encoded with a corresponding said LRU representation from the LRU table.